

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
Project MAC

Artificial Intelligence Project
Memo 62--

Memorandum MAC-M-124
December 12, 1963

Derivator I

A Program for Visual Inspection of Solutions to First-Order Non-Linear
Differential Equations.

by M. L. Minsky

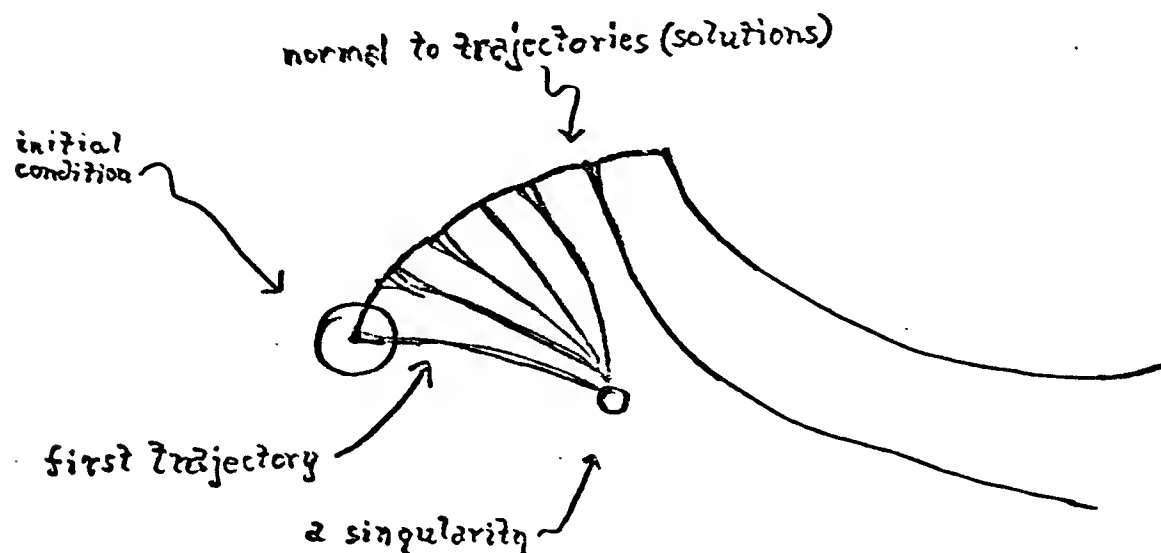
Derivator is a PDP-1 program for examining the solutions to differential equations of the form

$$\frac{dy}{dx} = f(x,y)$$

by inspection of a visual display of the trajectories. Because fixed-point arithmetic is used (in order to maintain visual display speeds), Derivator must be regarded as a qualitative tool. It is subject to truncation error in the trajectory-following program, and round-off error due to "underflow" in the function-definition programs for dy and dx. Still, it appears to be very suitable for studying the topology of solutions around singularities, etc. The display shows the solution curves ("characteristics") in the x-y plane. They are generated parametrically.

Derivator's controls

Derivator is easy to use. It is controlled by sense-switches, the light-pen, and by program parameters that can be typed-in. When the program is started, the situation looks like this:



Somewhere in the field there is a small bright circle; this surrounds the "initial condition point" (q_x, q_y) . This circle can be moved freely using the light pen.

The computer starts integrating the differential equation starting at the point (q_x, q_y) . It continues this solution until either of

- 1) The length of the trajectory exceeds the value of the program parameter arc (in approximate centimeters). This is initially set at 0 set at one-half screen size.
- 2) The trajectory follows over the edge of the scope ((x, y) goes out of the unit square).
- 3) There is an overflow in evaluating the function definition.
- 4) The trajectory encounters what the program thinks is a singularity is suspected when both dy and dx are very small, that is when ds = $\sqrt{dx^2 + dy^2}$ is less than eps, a program parameter. This is presumptive evidence that $dy/dx = 0/0$ and hence is undefined.

As soon as the first trajectory is completed (by one of the above conditions) the display generates a short line-segment along the normal to the field. Then the process is repeated, generating another, more-or-less parallel curve, also a trajectory. This whole process is repeated several times, generating a family of solutions. The number of trajectories is bounded by gcc, a program parameter (of the form "law i n"), initially set at (octal) 20. However the process terminates also if any of the above conditions occurs along the normal path that ^{generates} ~~generates~~ the family of curves.

Whenever a continuation encounters a (presumed) singularity, the

trajectory is discontinued and terminated by a small bright circle, about 1/2 cm. in diameter.

The light-pen tracking program associated with the initial-condition-point can be used by the operator to explore the field, in case the initial display doesn't adequately show the interesting features of the field. If no trajectories appear, but only the bright circle, this is because the initial point is located in a badly-defined region--that is, one in which an overflow condition occurs. Move it around.

Note: The light-pen causes a re-initialization of the solution program. No solutions are shown while the pen is tracking. To release tracking, just snap the pen away quickly.

Switch features

The six switches have special functions.

Switch 1 (up) causes the direction field to be displayed, between solution displays. To an extent, this lets one get a picture of the whole field. The direction field is indicated by vectors on lattice points. (The sign of the vectors is not indicated.) Where parts of the field appear to be missing, overflow in the derivative functions is occurring.

The field displayed by Switch 1 is light-pen sensitive! When a point in this field is captured, as indicated by appearance of a small circle and cessation of other features, that position is recorded internally (in (px,py)). See below.

Note: This light-pen feature conflicts with the initial-condition-tracker, and should be turned off when necessary.

Switch 2 is a magnifier feature. When switch 2 is up, the field is

translated to have its origin at the point (px,py) selected by the light-pen field of Switch 1, and the plane is dilated by a scale factor of 2. This helps to examine the fine-structure around a singularity. (Further magnification can be obtained only by suitably adjusting the sar 1 and sal 1 instructions in the program ~~associated with the~~ ^{associated with the} ~~sas i 20 instructions.~~ ^{sas i 20 instructions.}

Switch 3. When Switch 3 is up, a new display, "Expensive Protractor" appears, and controls a rotation matrix applied to the vector (dy,dx). Every vector in the entire tangent field is rotated by the amount indicated by the angular pointer that occasionally flashes from the origin to the peripheral circular scale. The circular scale is light-pen-sensitive. To change the angular rotation matrix, capture the desired peripheral point with the light pen, and release it radially. Four poles are distinguished by little semi-circles. The "zero" angle is (unfortunately) at the top of the display. If the "east" or "west" pole is captured, then the vector field is rotated by a right angle, and in the trajectory display, equipotentials are exchanged with gradients. By flipping Switch 3 up and down, accordingly, one can see the orthogonal families that form a locally Cartesian mesh.

The components of the rotation matrix are stored in r1 and r2.

Sw Switch 4. Switch 4 converts the trajectory family from a single, one-sided family of curves to a double, two-sided family, thus, giving a more comprehensive picture of the situation. Since the display cycle is therefore four times ^{longer} ~~greater~~, this was made an option.

Switches 5 and 6. These are used to select the method of integration, and are normally left down. Putting ~~Switches~~ up makes things much worse, for it replaces the second-order method by a simple first-order tangent-following approximation. This is useful only to make the display run faster (about 2x), but the results cannot be trusted because of the large truncation error. Switch 5 up puts in another second-order method, only slightly different from the normal one.

The Test-Word

The test-word can be used as a convenient means for introducing a parameter into your function definitions. We have used the right-most bit of the test-word as an additional switch, however. When this switch is up, the trajectory display (and all others) is replaced by a scintillating display of short, randomly selected, solutions. In a well-darkened room, this gives a really spectacular picture of the general behavior of the solutions of the equation.

Some useful internal parameters

The basic integration-step is controlled by a set of shift operations executed from locations hr, hl, gr, gl. To halve the step-size (and hence more-than-double the precision) change hr from sar 71 to sar 73. One can go further, but it is necessary to make similar and complementary changes in the other three, lest indexing underflows kill the display.

gcc controls the number of trajectories.

arc is the length of an unobstructed trajectory.

In this program, we use $\sqrt{dx^2 + dy^2} = ds$. This metric is convenient for computation, and in some respects is better than the Pythagorean distance

measure, since it better fits the unit-square domain! It does mean that the true arc length of a trajectory depends somewhat on the curve's path, though for almost parallel curves this is not a significant distraction.

To write a function

Derivator does not yet have its little algebraic compiler, so one has to write in machine language. This is quite easy, however, for a copy of DDT is included at load-time and you can write simple functions with only a few minutes' instruction.

The functions are really defined by writing two programs, one for

$$\frac{dy}{ds} = f(x,y)$$

and one for

$$\frac{dx}{ds} = g(x,y).$$

The programs for these functions are written, starting at register "f" (=2000) and must end with a jump instruction "jmp fr". To plot, for example, a circle one must have

$$\frac{dy}{dx} = -\frac{x}{y}$$

and one can get this, for example, by using

$$dy = x \quad \text{and} \quad dx = -y$$

as computed by the program

f/	lac x	. . .	load accumulator from x
f+1/	dac dy		deposit accumulator in dy
f+2/	lac y		load accumulator from y
f+3/	cma		complement accumulator
f+4/	dac dx		deposit accumulator in dx
f+5/	jmp fr		return to main program.

The "variables" x, y, dx, and dy, are already understood by the program,

and are all you need to refer to. If a function uses constants or temporary storage, we have provided some;

c, c1, c2, ..., c7; t, t1, t2, ..., t7.

Example: to compute the function

$$\frac{dy}{dx} = \frac{-x + y(r^2 - 1/2)^2}{y + x(r^2 - 1/2)^2}$$

one can write

```
f/      lac x
        mul x
        dac t
        lac y
        mul y
        add t
        sub c
        dac t
        lac y
        mul t
        sub x
        dac dy
        lac x
        mul t
        add y
        dac dx
        jmp fr
```

provided that one puts $1/2 = 200000_8$ in register c to begin with. To set a constant, c_i , from the test word, use the instruction pair

```
lat      ("Load accumulator from test word")
dac di  ("Deposit accumulator in ci")
```

You have to know more about the PDP-1 to arrange proper scaling, and to use division.

NOTE: This is not a final version.